

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-14341

(P2001-14341A)

(43) 公開日 平成13年1月19日 (2001.1.19)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 17/30

識別記号

F I

G 0 6 F 15/401  
15/40

テーマコード\* (参考)

3 1 0 A 5 B 0 7 5  
3 7 0 A

審査請求 未請求 請求項の数12 O L (全 16 頁)

(21) 出願番号 特願平11-188613

(22) 出願日 平成11年7月2日 (1999.7.2)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 石岡 恒憲

東京都目黒区駒場2-19-23 大学入試センター内

(72) 発明者 亀田 雅之

東京都大田区中馬込1丁目3番6号 株式会社リコー内

(74) 代理人 100101177

弁理士 柏木 慎史 (外1名)

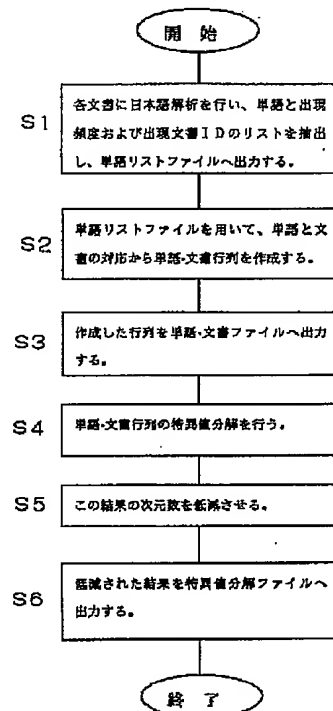
Fターム(参考) 5B075 ND03 NK02 NK32 PQ05 PQ46  
PR10 UU40

(54) 【発明の名称】 データベース作成装置および関連文書／関連語検索装置、データベース作成方法および関連文書／関連語検索方法ならびに記憶媒体

(57) 【要約】

【課題】 小さなメモリ空間でも関連文書／関連語の検索を行うことができるようにする。

【解決手段】 文書群ファイルに含まれた各文書に対して自然言語解析を行い、出現単語、この出現単語の出現頻度、およびその単語が出現した文書のIDのリストを抽出し、単語リストファイルに保存する (ステップS1)。次に、この単語リストファイル中の出現単語による情報から、行方向に出現単語、列方向に文書番号をとって、単語一文書行列を作成する (ステップS2)。そして、単語一文書行列を特異値分解して、特異値ベクトル、単語行列、文書行列を得 (ステップS4)、この特異値ベクトル、単語行列、文書行列の各々について要素を大きい方からk個だけ取り出し、新たな特異値ベクトル、単語行列、文書行列を生成することで次元数を低減する (ステップS5)。



## 【特許請求の範囲】

【請求項 1】 文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する単語抽出部と、

前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する単語一文書対応作成部と、

この単語一文書行列を特異値分解する特異値分解部と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する次数低減部とを備えている関連文書／関連語検索用のデータベース作成装置。

【請求項 2】 前記単語一文書対応作成部は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである請求項 1 に記載のデータベース作成装置。

【請求項 3】 請求項 1 または 2 に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群の各文書との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する関連文書検索部とを備えている関連文書検索装置。

【請求項 4】 請求項 1 または 2 に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する関連語検索部とを備えている関連語検索装置。

【請求項 5】 文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する工程と、

この単語一文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とを含んでなる関連文書／関連語検索用のデータベース作成方法。

【請求項 6】 前記単語一文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである請求項 5 に記載のデータベース作成方法。

【請求項 7】 請求項 5 または 6 に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連

が深い順に前記文書群から検索する工程を含んでなる関連文書検索方法。

【請求項 8】 請求項 5 または 6 に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する工程を含んでなる関連語検索方法。

10 【請求項 9】 文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、

前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する工程と、

この単語一文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とにより、関連文書／関連語検索用のデータベースの作成をコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

20 【請求項 10】 前記単語一文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである請求項 9 に記載の記憶媒体。

30 【請求項 11】 請求項 9 または 10 に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

【請求項 12】 請求項 9 または 10 に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

## 40 【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、データベース作成装置および関連文書／関連語検索装置、データベース作成方法および関連文書／関連語検索方法ならびに記憶媒体に関する。

【0002】

50 【従来の技術】近年、急速に関心の高まってきているデータマイニングの分野において、その適用分野の一つである文書マイニングは、インターネットのホームページを検索する検索エンジン利用の普及と伴って、コンピュ

## 3

ータの非専門家にとってもとりわけ注目の高いところである。

【0003】文書マイニングでは、扱うデータ量が膨大である上に、実用に耐えうる速度での応答が求められるため、その解決方法の一つとして、我々は単語の共起に基づいた検索アルゴリズムに注目してきた。「単語の共起」とは、同一の文書／文に複数の単語が同時に出現することをいう。

【0004】従来、単語の共起に着目した「文書マイニング」には大別して2つのアプローチがあったと思われる。一つは、入力キーワードを含む文書集合に成立する相関ルールを求め、そのルールに基づき検索をおこなうものである。発見すべき知識は、どのような単語同士が共起しやすいかである。

【0005】もう一つのアプローチは、入力キーワード／問い合わせ文と検索対象文書に現われる単語との共起の度合いによって、より適切と考えられる文書を検索するものである。基本的な考え方は、検索要求ベクトルに類似したベクトルをもつ文書は、適切な文書であると判断するもので、一般にはベクトル空間モデル (vector-space model) と呼ばれる。

【0006】その際に、単語の重み付けがしばしば行なわれるが、その方法として、単一文書中で出現する頻度 (within-story term frequency) に応じて重みを与えるtf法と、その単語が出現する文書数の逆数 (inverse document frequency) に応じて重みを与える (すなわち、さまざまな文書に出現するありふれた単語の重みを低くする) idf法とを組み合わせたtf・idfモデル、もしくはその派生が用いられることが多い。

【0007】統計的色彩が強い方法としては、Deerwesterによって提案された“Latent Semantic analysis”がある (“Deerwester, S., Dumais, S. T., G. W. Landauer, T. K. and Harshman, R. (1990): Indexing by latent semantic analysis. Journal of the American Society for Information Science, Vol. 41, No. 7, PP. 391-407” 参照)。これは、共起の頻度を示す単語一文書行列を特異値分解 (たとえば、“柳井晴夫、竹内啓(1983): 射影行列・一般逆行列・特異値分解、UP応用数学選書10、東京大学出版会” 参照) することにより、文書の潜在的意味構造を抽出するものである。得られた意味空間において、互いに関連した文書や単語は近接するように構成される。この方法も結果的にはベクトル空間モデルの一つであるが、共起という一種のパターンマッチを間接的に用いているために、「入力キーワードを全く含まないが意味的に近い」文書をも選ぶこともできるようになる。たとえば、“結婚”という語を入力キーワードにして、“結婚”という語を含まないけれども、“披露宴”や“新婚旅行”といった「結婚」に関連の深い単語を含む文書」を検索することが可能となる。

【0008】

## 4

【発明が解決しようとする課題】しかし、一般に特異値分解は巨大なメモリ空間を必要とし、データ数が数千を越える程度の大きさの問題に対してさえ、計算量の制限からワークステーションやパソコンの性能では実行不可能である。

【0009】この発明は、小さなメモリ空間でも関連文書／関連語の検索を行うことができるようにすることにある。

【0010】

10 【課題を解決するための手段】請求項1に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する単語抽出部と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する単語一文書対応作成部と、この単語一文書行列を特異値分解する特異値分解部と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する次数低減部とを備えている関連文書／関連語検索用のデータベース作成装置である。

20 【0011】したがって、単語一文書行列が疎 (大半の行列要素が0) であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

30 【0012】請求項2に記載の発明は、請求項1に記載のデータベース作成装置において、前記単語一文書対応作成部は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである。

【0013】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

40 【0014】請求項3に記載の発明は、請求項1または2に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群の各文書との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する関連文書検索部とを備えている関連文書検索装置である。

【0015】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

50 【0016】請求項4に記載の発明は、請求項1または2に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を前記デー

タを用いて測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する関連語検索部とを備えている関連語検索装置である。

【0017】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【0018】請求項5に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する工程と、この単語一文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とを含んでなる関連文書／関連語検索用のデータベース作成方法である。

【0019】したがって、単語一文書行列が疎（大半の行列要素が0）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【0020】請求項6に記載の発明は、請求項5に記載のデータベース作成方法において、前記単語一文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである。

【0021】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【0022】請求項7に記載の発明は、請求項5または6に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する工程を含んでなる関連文書検索方法である。

【0023】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【0024】請求項8に記載の発明は、請求項5または6に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する工程を含んでなる関連語検索方法である。

【0025】したがって、小さな記憶容量の関連文書／

関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【0026】請求項9に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語一文書行列を生成する工程と、この単語一文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とにより、関連文書／関連語検索用のデータベースの作成をコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【0027】したがって、単語一文書行列が疎（大半の行列要素が0）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【0028】請求項10に記載の発明は、請求項9に記載の記憶媒体において、前記単語一文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語一文書行列を生成するものである。

【0029】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【0030】請求項11に記載の発明は、請求項9または10に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【0031】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【0032】請求項12に記載の発明は、請求項9または10に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【0033】したがって、小さな記憶容量の関連文書／

関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【0034】

【発明の実施の形態】 (1) 本システムの理論

【特異値分解のもととなるデータ：】単語-文書行列Xを行方向に（抽出された）単語のリスト、列方向に各文書を取り、各文書毎に各単語が出現するかどうかを記憶

$$x_{ij}=0 \text{ (単語 } i \text{ が文書 } j \text{ に含まれないとき)} \quad \dots\dots (1)$$

とする。

【0036】【特異値分解：】行列Xを以下のように分

$$X = T_0 S_0 D_0^{\sim} \quad \dots\dots (2)$$

ここで、 $T_0$ は $t \times m$ 行列、 $S_0$ は $m \times m$ の正方対角行列（対角要素以外はすべて0）、 $D_0^{\sim}$ は $m \times d$ 行列である。また $0 \leq d \leq t$ とする。“ $\sim$ （ダッシュ）”は $D_0$ の転置を示す。 $S_0$ の対角要素は大きい順とする。

【0038】【疎行列（Sparse matrix）に適した特異値分解：】単語-文書行列は一般に巨大な疎行列となる。このような巨大な疎行列に対する特異値問題を解くために、最も単純なアルゴリズムである部分空間反復（サブスペース繰返し）法を用いる。この方法は古典的なべき乗法（power method）のブロック化と見なすことができる。この部分空間反復法にはいくつかのバリエーションがあるが、その内の一つは、“ $B = X^{\sim} X$ ”に対して次の式を更新する。

$$【0039】Z_i = B^i Z_0$$

ここで、“ $Z_0 = [z_1, z_2, \dots, z_s]$ ”は“ $d \times s$ ”である（“ $s$ ”は、計算上の精度を高めるために必要な特異値の数に余裕を加えた次元数。例えば、必要な特異値の数が50なら10を加えて60とする。以下同じ。）。 $i$ は、求めるBの特異値の数 $p$ まで繰り返す。ここで、列ベクトル“ $z_j$ （ $1 \leq j \leq s$ ）”は、“ $\|z_j\| = 1$ ”で、かつ、互いに独立になるように適当に定める。このようにすれば、列ベクトル $z_j$ は行列Bの主要な特異値に収束してゆく。これにより、行列 $Z_i$ は漸次、

$$X^{\sim} \text{ (Xのハット)} = T S D^{\sim} \quad \dots\dots (3)$$

として作成される行列 $X^{\sim}$ は行列Xの近似となる。ここで行列Tは $t \times k$ 行列、行列Sは $k \times k$ の正方対角行列、行列 $D^{\sim}$ は $k \times d$ 行列である。本発明の実施の形態の対象とする言語データのような場合、経験的に $k$ は50～100程度にするとよい。

【0042】【関連文書の検索：】ユーザからの問い合わせを擬似文書 $q$ と考える。例えば、複数の単語の組み合わせ等から問い合わせ文を作る。この擬似文書 $q$ から単語を抽出して、 $t$ 次元の単語ベクトル $x_q$ で表現することができる。これを用いて、文書行列Dの行に対応する $1 \times k$ の文書ベクトル

$$d_q = x_q^{\sim} T S^{-1}$$

$$r(d_q, d_c) = (d_q, d_c) / \|d_q\| \|d_c\| \quad \dots\dots (4)$$

ここで $d_c$ は、単語-文書行列Xの $c$ 列を表している、 $t$ 次元の単語ベクトル $x_c$ を使って“ $d_c = x_c^{\sim} T$ ”

させたものと定義し、次のように表現する。

【0035】単語-文書行列Xは、 $t$ を単語数、 $d$ を文書数としたとき $t \times d$ 行列で表し、その各要素を $x_{ij}$ としたとき、

$x_{ij} = 1$ （単語 $i$ （ $1 \leq i \leq t$ ）が文書（ $1 \leq j \leq d$ ）に含まれるとき）

または

解する。

【0037】

列ごとの線形独立性を失ってゆく。そこで行列Bの大きな $p$ 個の特異値ペアを近似するために、各ステップにおいて修正Gram-Schmidtプロシージャを用い、 $z_j$ を互いに直交にすれば、それらの間の線形独立性が保たれることをBauerは示した（“Bauer, F. L. (1957年): Das Verfahren der Treppeniteration und verwandte zur Lösung algebraischer Eigenwertprobleme, ZAMP, 8, 214-235”参照）。

【0040】しかしながら、 $z_j$ のBの特異値ベクトルに対する収束速度はわずか1次に過ぎない。そこで部分空間反復に、洗練されたRutishauser (1970年) のritzit プログラム（部分空間反復に、さらにRayleigh-RitzプロシージャとChebyshev多項式を経た高速化を行なっている）を使用している（“Rutishauser, H. (1970年): On the rates of convergence of the Lanczos and the block-Lanczos methods, SIAM Journal of Numerical Analysis, vol.17, pp687-706”参照）。

【0041】【次元低減：】上記行列Xの特異値分解で得た行列 $S_0$ の対角要素のうち大きいほうから $k$ 番目までを取り、これを新たな正方対角行列Sとする。それに対応して、行列 $T_0$ および行列 $D_0$ も $k$ 列までを抜き出し、これを新たな行列TおよびDとする。このとき、

を導くことができる。

【0043】ここでTは $t \times k$ 行列、Sは $k \times k$ 正方対角行列である。“ $\sim$ （ダッシュ）”は転置を、“ $^{-1}$ ”は逆行列を示し、“ $S = \text{diag}(\mu_1, \mu_2, \dots, \mu_k)$ ”としたとき、“ $S^{-1} = \text{diag}(1/\mu_1, 1/\mu_2, \dots, 1/\mu_k)$ ”である。このとき擬似文書 $q$ の文書ベクトル $d_q$ （ $k$ 次元ベクトル）に対し、比較の対象とする文書 $c$ の文書ベクトルを $d_c$ （ $k$ 次元ベクトル）とすれば、両文書の相関係数 $r(d_q, d_c)$ は、両文書がなす角の余弦で与えられる。

【0044】

$S^{-1}$ で求められる。これより、擬似文書 $d_q$ に近い文書を、近さの順に提示することが可能となる。なお

(4) 式の右辺分子の括弧は、内積を示す。

【0045】[関連語の検索:] 擬似文書 $q$ は、 $t$ 次元の単語ベクトル $x_q$ で表現することにより、擬似文書の

$$t_q = (\text{擬似文書 } q \text{ に含まれている単語に対応する行列 } T \text{ の行ベクトル } t_i \text{ の平均}) \quad \dots\dots (5)$$

に定めることができる。このようにすれば、比較の対象とする単語 $c$ に対する行列 $T$ の $c$ 行で表される $k$ 次元ベ

$$r(t_q, t_c) = (t_q, t_c) / \|t_q\| \|t_c\| \quad \dots\dots (6)$$

として与えられる。これより擬似文書 $q$ に関係の深い単語をその近さの順で提示することができるようになる。

【0046】(2) 各システムの動作環境

本システムは、システム単体でもネットワーク環境下でも動作する。典型的には、図1に示すように、クライアント2と、データベース作成装置、関連文書/関連語検索装置であるサーバ3とがネットワーク4で接続されているクライアント/サーバシステム1で構成され、データベース5を格納するサーバ3に対して、クライアント2から検索要求を問い合わせ、その検索結果をサーバ3がクライアント2に返すものである。

【0047】図2は、クライアント2、サーバ3として用いるコンピュータの概略構成を示すブロック図である。図2に示すように、このコンピュータ2、3は、CPU6と、ROM、RAMなどの記憶装置7とがバス8で接続されている。また、バス8には、記憶媒体であるCD-ROM9を読み取るCD-ROMドライブ10を制御するためのCD-ROMドライブ制御部11と、CRTなどの表示装置12およびキーボード、マウスなどの入力装置13を制御する入出力制御部14と、ハードディスク15を制御するハードディスク制御部16と、コンピュータ2、3をネットワーク4と接続するためのLAN制御部17とが接続されている。クライアント2のハードディスク15には、CD-ROM9から本システムのクライアント用ソフトがインストールされ、サーバ3のハードディスク15には、CD-ROM9から本システムのサーバ用ソフトがインストールされている。

【0048】(3) 本システムで利用するデータ構造

(A) 本システムで使用する各種ファイル

本システムでは、以下の種類のデータ(図3参照)をファイルとしてサーバ3のハードディスク15へ保持し、検索時に利用する。

- ・検索対象となる文書群を保持する文書群ファイル21
- ・文書群ファイルから抽出した単語を保持する単語リストファイル22
- ・文書群ファイルから抽出された単語が各文書で存在するかどうかを示す行列(単語一文書行列 $X$ )を保持する単語一文書ファイル23
- ・単語一文書ファイルから特異値分解をした結果を保持する特異値分解ファイル24

【0049】(B) 各種ファイルのデータ構造

- ・文書群ファイル21(図3(a)参照)

座標をその擬似文書が含む単語群の中心(centroid)、すなわち、

クトルを $t_c$ とすると、両単語の相関係数は、(4)式と同じように、

各文書毎に、表題、文書の種類、書誌事項(著者名、出版社名、発行日等)、要約文等から構成される。

【0050】・単語リストファイル22(図3(b)参照)

各単語毎に、単語の表記、この単語が文書群中に出現する頻度、および、この単語が出現した文書IDのリストで構成される。

【0051】・単語一文書ファイル23(図3(c)参照)

次のような順序で構成する。行列の行数、行列の列数、行列中の非ゼロ要素数、列ごとに読み込んだ場合の各列先頭時点における非ゼロ要素の累積数+1、各列における非ゼロ要素の行番号、非ゼロ要素の値そのもの(整数、あるいは実数)

【0052】・特異値分解ファイル24(図3(c)参照)

このファイルには、次の3つが含まれる。特異値のリスト、特異値分解された単語行列 $T$ 、特異値分解された文書行列 $D$

【0053】(4) 本システムの処理

(A) 全体の流れ

本システムの検索を行うためには、まず、文書群に対して予め文書群で使用している単語の抽出を行い、各文書がこれらの単語のうちどれを含んでいるかを示す単語一文書ファイル23を作成し、このファイルの特異値分解し、次元の低減を行った結果の特異値分解ファイル24に保持する。関連文書検索は、その結果を用いて、ユーザからの問い合わせに応じた文書を関連の度合いの大きい順に出力する。また、関連語検索も同様に解析結果を用いて、ユーザーからの問い合わせに応じた関連単語を関連の度合いの大きい順に出力する。

【0054】(B) 予備データの生成

以下では、図4に示すサーバ3の機能ブロック図、図5に示すサーバ3が行う処理のフローチャートに基づいて、本システムの予備データの生成の処理について説明する。

【0055】まず、単語抽出部25において、文書群ファイル21に含まれた各文書に対して自然言語解析を行い、出現単語、この出現単語の出現頻度、およびその単語が出現した文書のIDのリストを抽出し、単語リストファイル22に保存する(ステップS1)。

【0056】次に、単語一文書対応作成部26で、この

単語リストファイル 22 中の出現単語による情報から、行方向に出現単語、列方向に文書番号をとって、単語一文書行列 X を作成し (ステップ S2)、単語一文書ファイル 23 に出力する (ステップ S3)。

【0057】尚、大きな文書ほど単語の共起が起きやすいので、1 文書に現れる単語数で共起頻度を割り、さらに各文書 (各列) における要素の和が 1 となるように基準化する。この基準化は、関連文書検索において、大きな文書ほど関連文書として検索されやすいことへの対処である。また、基準化の方法としては、各文書 (各列) における要素の 2 乗和が 1 となるようにしてもよい。

【0058】下に示した例は、9 文書に出現する 12 単語の共起関係を現したものである (列方向に文書、行方向に単語をとってあり、[] で示した数字は行および列の番号を示している。)

【0059】

【表 1】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.33	0.00	0.00	0.33	0.00	0	0.0	0.00	0.00
[2,]	0.33	0.00	0.25	0.00	0.00	0	0.0	0.00	0.00
[3,]	0.33	0.17	0.00	0.00	0.00	0	0.0	0.00	0.00
[4,]	0.00	0.17	0.25	0.00	0.33	0	0.0	0.00	0.00
[5,]	0.00	0.17	0.25	0.33	0.00	0	0.0	0.00	0.00
[6,]	0.00	0.17	0.00	0.00	0.33	0	0.0	0.00	0.00
[7,]	0.00	0.17	0.00	0.00	0.33	0	0.0	0.00	0.00
[8,]	0.00	0.00	0.25	0.33	0.00	0	0.0	0.00	0.00
[9,]	0.00	0.17	0.00	0.00	0.00	0	0.0	0.00	0.33
[10,]	0.00	0.00	0.00	0.00	0.00	1	0.5	0.33	0.00
[11,]	0.00	0.00	0.00	0.00	0.00	0	0.5	0.33	0.33
[12,]	0.00	0.00	0.00	0.00	0.00	0	0.0	0.33	0.33

【0060】このような疎行列である単語一文書行列 X をそのままハードディスク 15 に格納すると膨大な領域を必要とするので、下記のように、Harwell-Boeing sparse matrix format で格納することによって ( "Duff, et

al. (1989年): Sparse Matrix Test problems, ACM TOMS (Transaction on Mathematical Software) Vol.15, No.1, March 1989" 参照)、記憶容量の節約、ならびにデータ読み込み時間の大幅な低減をはかることができる。

【0061】例えば、上述の単語一文書行列 X に対しては、以下の形式で単語一文書ファイル 23 に格納される (なお、見やすくするために改行を入れて示す)。

12 9 28 ← 行列の行数、列数、非ゼロ要素数

1 4 10 14 17 20 21 23 26 29 ← 各列先頭時点における非ゼロ要素の累積数 + 1

1 2 3 ← 各列における非ゼロ要素の行番号、ここから

3 4 5 6 7 9

2 4 5 8

1 5 8

4 6 7

10

10 11

10 11 12

9 11 12 ← 各列における非ゼロ要素の行番号、ここから

20 0.33 0.33 0.33 ← 非ゼロ要素の値そのもの、ここから

0.17 0.17 0.17 0.17 0.17 0.17

0.25 0.25 0.25 0.25

0.33 0.33 0.33

1

0.5 0.5

0.33 0.33 0.33

0.33 0.33 0.33 ← 非ゼロ要素の値そのもの、ここから

【0062】次に、特異値分解部 27 で、単語一文書行列 X を式 (2) のように特異値分解する (ステップ S4)。上記の X に対して、 $T_0$  は次のようになる ( $S_0$  は対角要素のみ示す)。

【0063】

【表 2】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.000	-0.398	0.112	-0.404	0.223	0.146	-0.541	0.253	-0.320
[2,]	0.000	-0.338	0.083	-0.212	0.387	-0.322	0.556	0.074	-0.234
[3,]	-0.001	-0.258	0.034	-0.106	0.584	0.112	-0.023	-0.264	0.524
[4,]	-0.001	-0.397	0.011	0.470	-0.064	-0.220	0.300	0.063	-0.079
[5,]	-0.001	-0.464	0.081	-0.130	-0.447	0.123	0.018	-0.276	0.498
[6,]	-0.001	-0.236	-0.023	0.484	0.096	0.040	-0.265	0.102	-0.021
[7,]	-0.001	-0.236	-0.023	0.484	0.096	0.040	-0.265	0.102	-0.021
[8,]	0.000	-0.383	0.096	-0.221	-0.484	-0.052	0.033	0.101	-0.202
[9,]	-0.039	-0.130	-0.347	0.054	0.040	0.588	0.185	-0.519	-0.455
[10,]	-0.925	0.048	0.349	0.051	0.009	0.122	0.051	0.005	-0.022
[11,]	-0.351	-0.079	-0.675	-0.115	-0.025	-0.537	-0.254	-0.217	0.003
[12,]	-0.140	-0.067	-0.513	-0.075	-0.008	0.380	0.253	0.657	0.257

【0064】上記の X に対して、 $S_0$  は次のようになる。 ( $S_0$  は対角要素のみ示す)

[1] 1.230 0.783 0.710 0.631 0.488 0.337 0.317 0.24

50 6 0.136

上記のXに対してD<sub>0</sub>は次のようになる。

【表 3】

【0065】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.000	-0.419	0.107	-0.378	0.808	-0.064	-0.008	0.084	-0.072
[2,]	-0.006	-0.374	-0.064	0.338	0.107	0.345	-0.028	-0.547	0.558
[3,]	-0.001	-0.505	0.096	-0.037	-0.311	-0.350	0.716	-0.038	-0.031
[4,]	0.000	-0.525	0.135	-0.395	-0.479	0.212	-0.511	0.105	-0.059
[5,]	-0.001	-0.366	-0.016	0.752	0.087	-0.137	-0.240	0.358	-0.296
[6,]	-0.752	0.061	0.492	0.082	0.018	0.361	0.160	0.020	-0.160
[7,]	-0.519	-0.020	-0.230	-0.050	-0.017	-0.617	-0.322	-0.431	-0.069
[8,]	-0.380	-0.041	-0.390	-0.072	-0.017	-0.035	0.052	0.596	0.580
[9,]	-0.142	-0.117	-0.714	-0.071	0.005	0.422	0.191	-0.106	-0.474

【0066】特異値分解によって求められた各行列は、文書数、あるいは単語数が膨大になるとT<sub>0</sub>、S<sub>0</sub>、D<sub>0</sub>を保持するためのメモリ空間も膨大なものとなるので、次数低減部28で次のようにして次数の低減を図り（ステップS5）、その結果を記憶部である特異値分解ファイル24に出力する（ステップS6）。

【0067】すなわち、特異値ベクトルS<sub>0</sub>は、単語—  
文書行列Xのrank（本例では9）の数だけ要素が存在するが、この特異値のベクトルの大きい方からk個だけ取り出し、これを新たにSと置く。これに対応し、行列T<sub>0</sub>、D<sub>0</sub>に対しても、それぞれ先頭からk列を取り出して新たにそれぞれ行列T、Dとおく。このようにして作成されたT、S、Dを保持することにより大幅な資源（ディスクやメモリなど）の節約を達成する。例えば、特異値のベクトルの大きい方からk=2個だけ取り出すとすると、行列Sは次のようになる（以下では説明の簡便性を考慮してk=2とするが、実際の大規模データに対しては経験的にk=50～100程度とすると良い）。

[1] 1.230 0.783

	[,1]	[,2]
[1,]	0.000	-0.419
[2,]	-0.006	-0.374
[3,]	-0.001	-0.505
[4,]	0.000	-0.525
[5,]	-0.001	-0.366
[6,]	-0.752	0.061
[7,]	-0.519	-0.020
[8,]	-0.380	-0.041
[9,]	-0.142	-0.117

【0071】このようにしたとき、“ $\hat{X}=TSD^{\sim}$ ”はXの近似となる。

上例の行列Tは、次のようになる。

【0068】

【表 4】

	[,1]	[,2]
[1,]	0.000	-0.398
[2,]	0.000	-0.338
[3,]	-0.001	-0.258
[4,]	-0.001	-0.397
[5,]	-0.001	-0.464
[6,]	-0.001	-0.236
[7,]	-0.001	-0.236
[8,]	0.000	-0.383
[9,]	-0.039	-0.130
[10,]	-0.925	0.048
[11,]	-0.351	-0.079
[12,]	-0.140	-0.067

【0069】上例の行列Dは、次のようになる。

【0070】

【表 5】

【0072】

【表 6】



15		[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	16
	[1,]	0.131	0.116	0.158	0.164	0.114	-0.019	0.006	0.013	0.036	
	[2,]	0.111	0.099	0.134	0.139	0.097	-0.016	0.005	0.011	0.031	
	[3,]	0.085	0.075	0.102	0.106	0.074	-0.011	0.005	0.009	0.024	
	[4,]	0.130	0.116	0.157	0.163	0.114	-0.018	0.007	0.013	0.036	
	[5,]	0.152	0.136	0.184	0.191	0.133	-0.021	0.008	0.016	0.043	
	[6,]	0.077	0.069	0.093	0.097	0.067	-0.010	0.004	0.008	0.022	
	[7,]	0.077	0.069	0.093	0.097	0.067	-0.010	0.004	0.008	0.022	
	[8,]	0.126	0.112	0.151	0.157	0.110	-0.018	0.006	0.013	0.035	
	[9,]	0.043	0.038	0.052	0.054	0.037	0.030	0.027	0.022	0.019	
	[10,]	-0.015	-0.007	-0.018	-0.019	-0.013	0.858	0.589	0.431	0.158	
	[11,]	0.026	0.026	0.032	0.033	0.023	0.321	0.225	0.167	0.069	
	[12,]	0.022	0.021	0.026	0.027	0.019	0.127	0.091	0.068	0.031	

### 【0073】(C) 関連文書検索

以下では、図6に示すサーバ3の機能ブロック図、図7に示すサーバ3が行う処理のフローチャートに基づいて、本システムの関連文書検索の処理について説明する。

【0074】クライアント2で入力した問い合わせを、問合わせ入力部29が受け付け（ステップS11）、受け取った文書検索文（以下、これを擬似文書 $q$ と呼ぶ）中にある単語（複数のときもある）を、関連文書検索部30が抽出し、この単語と文書群から抽出された単語リストファイル22とを照合して、 $t$ 次元の単語ベクトル $d_q = x_q^T S^{-1} = [0, -0.419]$

次に、この $d_q$ と検索対象である文書群の1つ1つに対して、両文書の相関係数を（4）式で計算する（ステップS14）。ここで各文書の文書ベクトル（文書 $c$ に対して文書ベクトルを $d_c$ と呼ぶ）は行列 $D$ の第 $c$ 行で与えられるから、例えば、対象文書を文書9とすると、 $d_c = [-0.142 \ -0.117]$

となる。これより、文書 $d_q$ と文書 $d_c$ との相関係数は、 $r(d_q, d_c) = \{0 \times (-0.142) + (-0.419) \times (-0.117)\} / [\sqrt{\{0^2 + (-0.419)^2\}} \cdot \sqrt{\{(-0.142)^2 + (-0.117)^2\}}] = 0.634$ となる。

【0078】このようにして、全文書に対する相関係数を求め、相関係数の大きい順に並べ替える。このうちの上位10番目までの相関係数を選び、この選び出された相関係数に対応する文書を、文書内容とともに、関連文書表示部31でクライアント2に表示させることができる（ステップS15）。

### 【0079】(D) 関連語検索

以下では、図8に示すサーバ3の機能ブロック図、図9に示すサーバ3が行う処理のフローチャートに基づいて、本システムの関連語検索の処理について説明する。

【0080】クライアント2からユーザが入力した問い合わせを、問合せ入力部29で受け取る（ステップS21）。この中から単語（複数も可能）を抽出する。この

$x_q$ を作成する（要素は基準化しておく）（ステップS12）。

【0075】例えば、擬似文書 $q$ が単語1,3を含んでいるとすれば、

$$x_q = [0.5, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0]$$

を作成する。

【0076】これを用いて、特異値分解ファイルから得た特異値ベクトル $S$ 、行列 $T$ 、 $D$ とから擬似文書 $q$ の文書ベクトル $d_q$ は（7）式のように計算することができる（ステップS13）。

【0077】

$$\dots\dots (7)$$

単語に対する行列 $T$ の行ベクトルを関連語検索部32で取り出す（ステップS22）。そして、この行ベクトルを抽出したすべての単語に対するこの行ベクトルを取り出して平均ベクトルを計算する（（5）式）。例えば、問い合わせに単語1,3を含んでいるとすれば次のようになる。

$$\text{【0081】 } t_q = [(0.0000.001)/2, (-0.398-0.258)/2] = [-0.001, -0.328]$$

次に、比較の対象となるすべての単語の単語ベクトル $t_c$ （ $k$ 次元ベクトル）に対し、両単語の相関係数を

（6）式で計算する（ステップS23）。この単語ベクトル $t_c$ は行列 $T$ の各行ベクトルに対応している。例えば、比較の対象とする単語を単語8とすると、 $t_c = [-0.000 \ -0.383]$

であるから、単語 $t_q$ と単語 $t_c$ との相関係数は次のように計算される。

$$\text{【0082】 } r(t_q, t_c) = \{(-0.001) \times (-0.000) + (-0.328) \times (-0.383)\} / [\sqrt{\{(-0.001)^2 + (-0.328)^2\}} \cdot \sqrt{\{(-0.000)^2 + (-0.383)^2\}}] = 0.958$$

このようにすべての相関係数を計算し、相関係数の大きい順に並び替え、大きい方から相関係数に対応した単語を表示することによって、問い合わせにある単語に關係の深い単語をその近さの順で提示することができる（ス

テップS24)。

【0083】(5) 実際の評価データ

ある全国紙の1年分(1994年)の記事から、先頭2, 055文書と、そこに出現する頻度4以上の4, 041単語を用いた場合、最もオーソドックスなハウスホルダー変換による特異値分解では、約12時間(60×60×12=43, 200秒)、所要メモリで約200MBが必要であるが、本システムを用いると(特異値の大きな方から50個を求めた場合)9.5秒、所要メモリで12.7MBを要した。

【0084】また20, 211文書と44, 883単語における特異値分解では、従来法では(メモリの制約から)計算不能であるが、本システムでは13.4秒、所要メモリで53.0MBで計算できた。

【0085】

【発明の効果】請求項1に記載の発明は、単語一文書行列が疎(大半の行列要素が0)であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書/関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書/関連語検索用のデータベースの記憶容量を低減することができる。

【0086】請求項2に記載の発明は、請求項1に記載のデータベース作成装置において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【0087】請求項3に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【0088】請求項4に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【0089】請求項5に記載の発明は、単語一文書行列が疎(大半の行列要素が0)であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書/関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書/関連語検索用のデータベースの記憶容量を低減することができる。

【0090】請求項6に記載の発明は、請求項5に記載のデータベース作成方法において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【0091】請求項7に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【0092】請求項8に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【0093】請求項9に記載の発明は、単語一文書行列が疎(大半の行列要素が0)であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書/関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書/関連語検索用のデータベースの記憶容量を低減することができる。

【0094】請求項10に記載の発明は、請求項9に記載の記憶媒体において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【0095】請求項11に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【0096】請求項12に記載の発明は、小さな記憶容量の関連文書/関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【図面の簡単な説明】

【図1】この発明の一実施の形態にかかるクライアント/サーバシステムの概略構成を示すブロック図である。

【図2】前記クライアントおよびサーバに用いるコンピュータの構成を説明するブロック図である。

【図3】前記サーバで用いる各種ファイルの構成を説明する図である。

【図4】前記サーバで行う予備データの生成の処理を説明する機能ブロック図である。

【図5】前記サーバで行う予備データの生成の処理を説明するフローチャートである。

【図6】前記サーバで行う関連文書検索の処理を説明する機能ブロック図である。

【図7】前記サーバで行う関連文書検索の処理を説明するフローチャートである。

【図8】前記サーバで行う関連語検索の処理を説明する機能ブロック図である。

【図9】前記サーバで行う関連語検索の処理を説明するフローチャートである。

【符号の説明】

3 データベース作成装置、関連文書/関連語検索装置

9 記憶媒体

24 記憶部

25 単語抽出部

26 単語一文書対応作成部

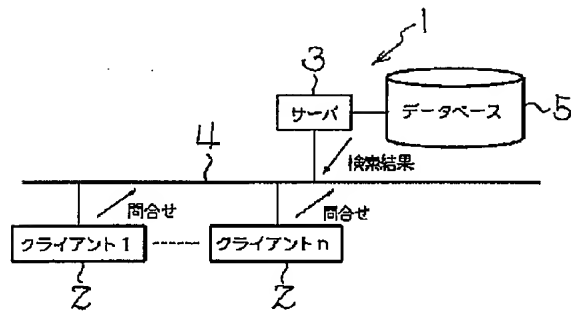
27 特異値分解部

50 28 次数低減部

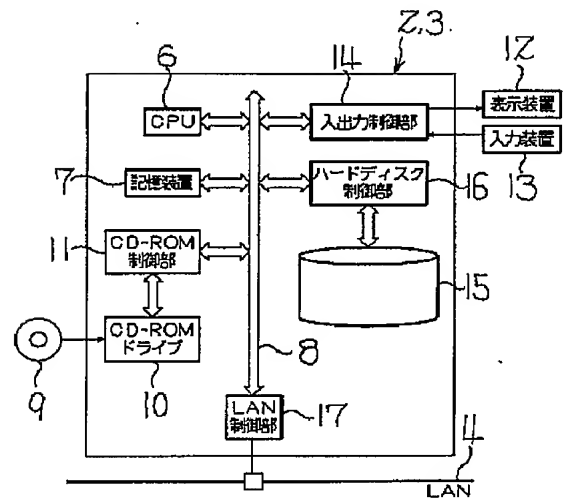
30 関連文書検索部

31 関連語検索部

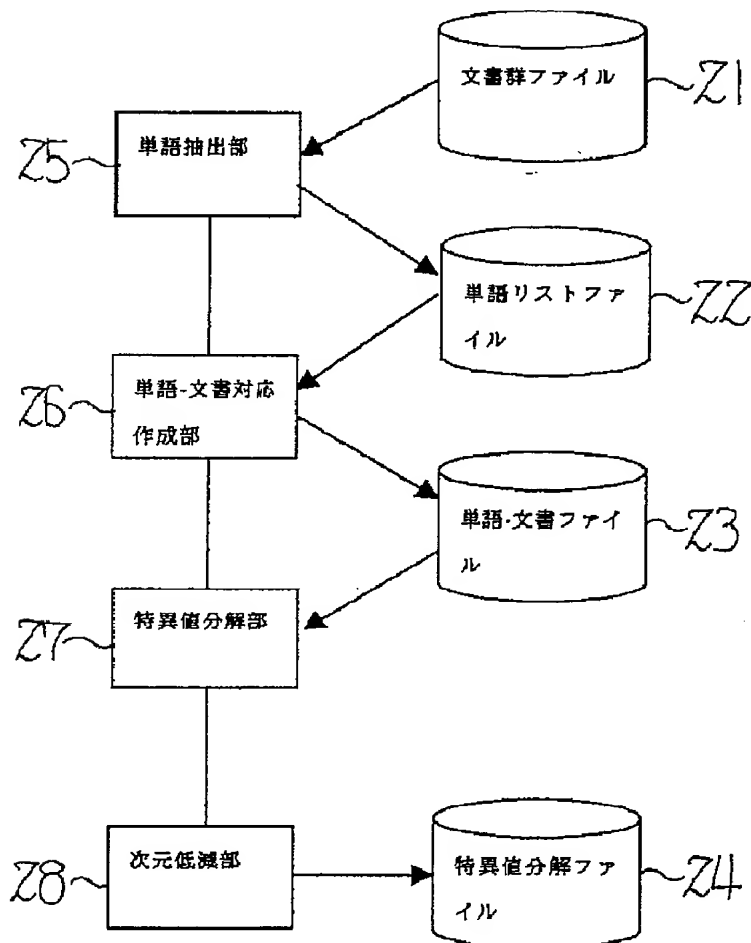
【図 1】



【図 2】



【図 4】



【図3】

## (a) 文書群ファイル

文書ID	表題	文書の種類	書誌事項	要約文
文書1				
文書2				
⋮				

## (b) 単語リストファイル

単語ID	単語表記	出現頻度	出現文書IDリスト
単語1			
単語2			
⋮			

## (c) 単語-文書ファイル

A	B	C	D <sub>1</sub>	.....	D <sub>2</sub>	E <sub>1</sub>	.....	E <sub>2</sub>	F <sub>1</sub>	.....	F <sub>2</sub>
---	---	---	----------------	-------	----------------	----------------	-------	----------------	----------------	-------	----------------

A: 行列の行数 (単語の数t)

B: 行列の列数 (文書の数d)

C: 行列中の非ゼロ要素の数

D<sub>i</sub>: (i-1) 列までの非ゼロ要素数の累積+1E<sub>i</sub>: i 列における非ゼロ要素がある行番号のリストF<sub>i</sub>: 非ゼロ要素の値 (第i列から順に並べる)

## (d) 特異値分解ファイル

$\lambda_1$	.....	$\lambda_k$
-------------	-------	-------------

 $\lambda_i$ : i 番目の特異値

(kは与えられた特異値の数)

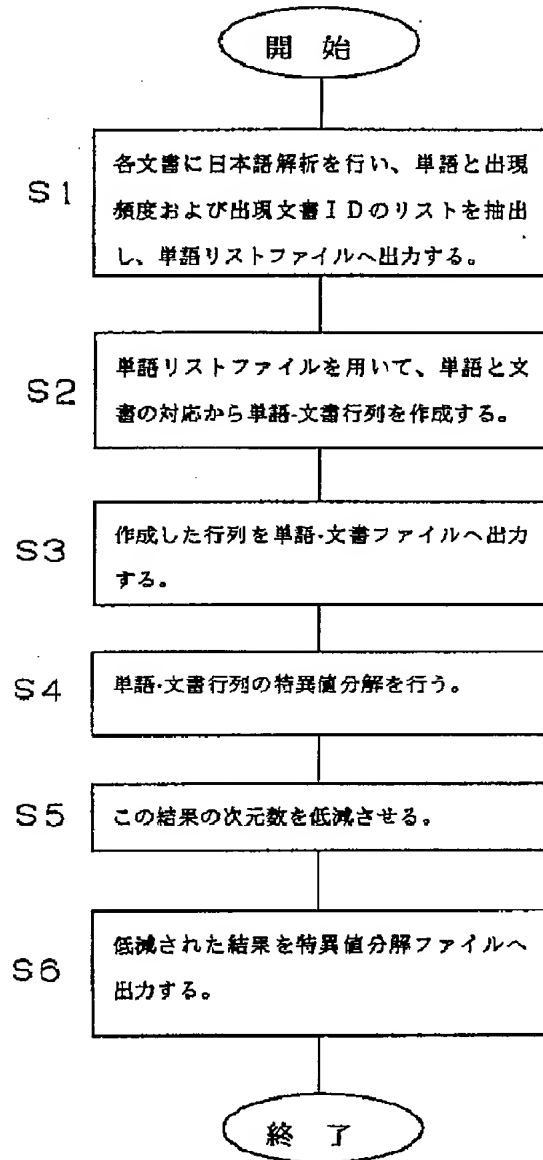
T <sub>11</sub>	.....	T <sub>1k</sub>
⋮		⋮
⋮	(行列T)	⋮
T <sub>t1</sub>	.....	T <sub>tk</sub>

t: 単語数

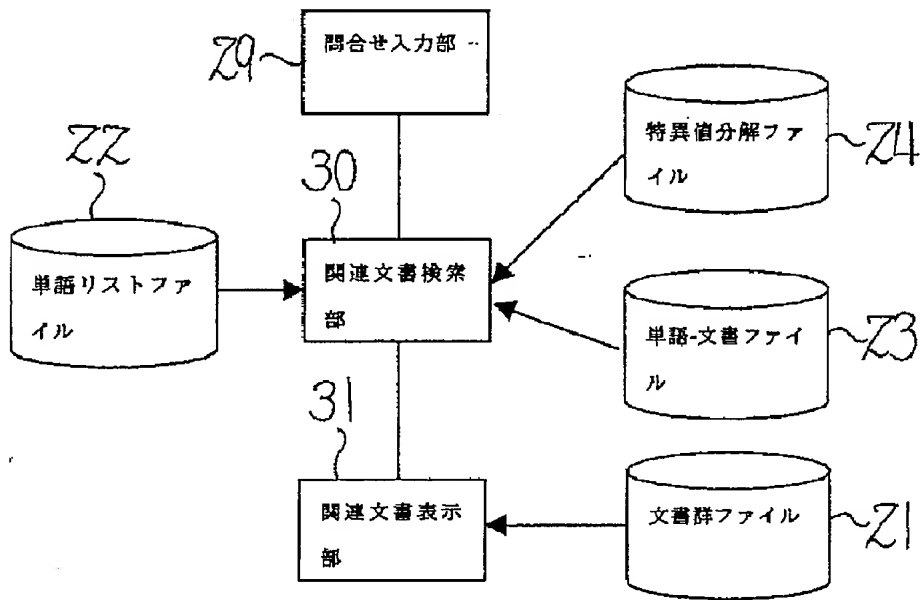
d: 文書数

D <sub>11</sub>	.....	D <sub>1k</sub>
⋮		⋮
⋮	(行列D)	⋮
D <sub>d1</sub>	.....	D <sub>dk</sub>

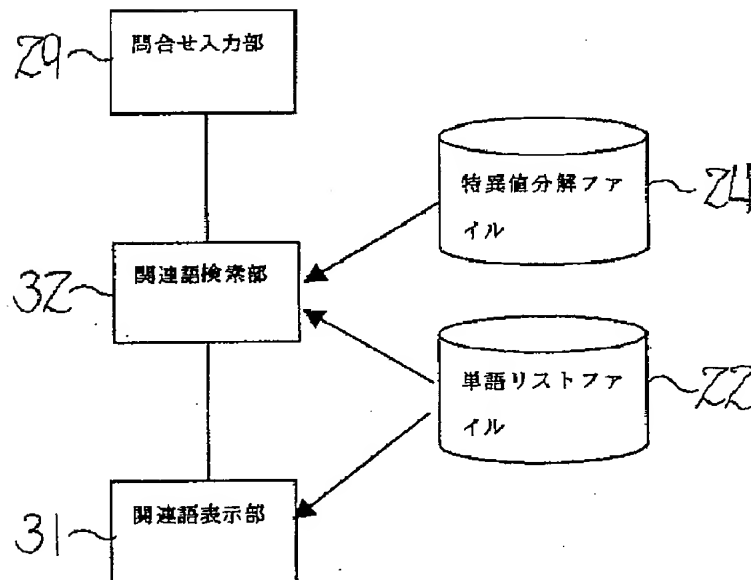
【図5】



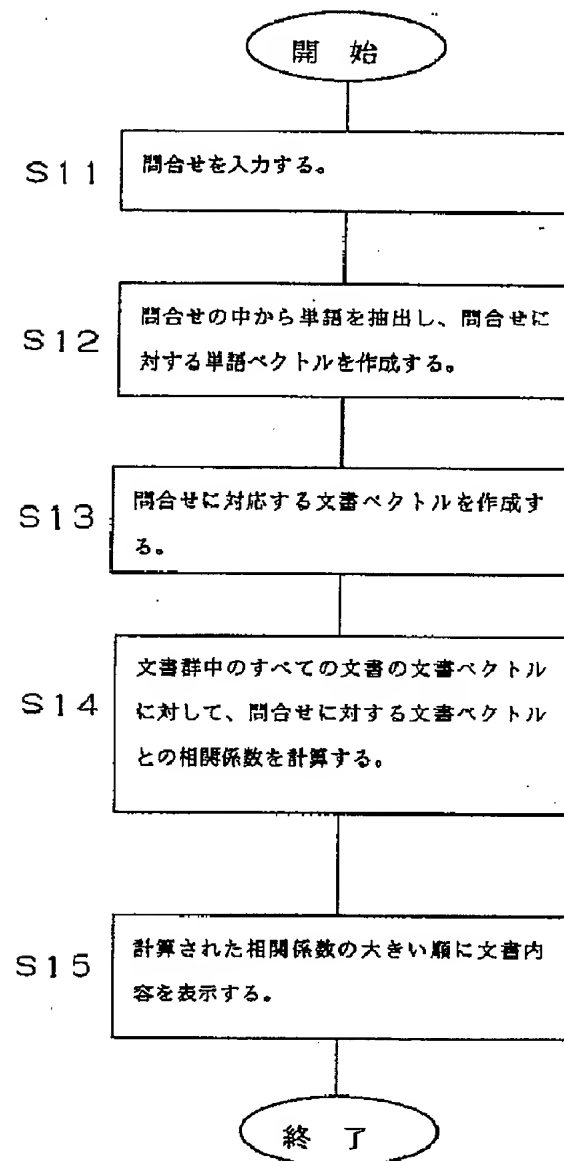
【図 6】



【図 8】



【図 7】



【図 9】

